R join_a=b S join_b=c T

R(a):

- 1000 tuples
- Primary tree index
- R.a uniform: 0-1000

S(b)

- 500 tuples
- Primary heap file
- Secondary hash index on b
- S.b uniform: 0-50

T(c)

- 10000 tuples
- Primary tree index
- Secondary tree index
- T.c uniform: 0-10
- ICARD(T.c) = 10

Tuples: 100bytes
Page: 1000 bytes
DE: 10 bytes
Fill factor: 100%
Tuples/page: 10
DE/page: 100

R(a) index:

- leaf/data pages: 1000(tuples) / 10 (tuples/page) = 100 pages
- Each directory page can point to 100 child pages
- 100 data pages / 100 de/page = 1 parent page of leaves
- Height = 1
- Selectivity of a=?: 1/1000
- R(a = ?): 1 directory page + 1 data page to read 1000*1/1000 tuples

S(b) secondary index:

- Selectivity(b=?): 1/50
- Cardinality(b=?): 500 / 50 = 10
- S(b=?): 500 tuples/100 (de/page) = 5 / 50 buckets = pages/bucket = 1 data page + # tuples = 11 pages.

T(c) primary index:

- Data pages = 1000
- Height = 2
- Selectivity of c=?: 1/10
- T(c=?): 2 dir pages + (1000*1/10) = 102 pages.

T(c) secondary index:

- Data pages = 10000 / 100 = 100
- Height: 1
- Cardinality of c=?: 10000 * selectivity = 1000
- T(c=?): 1 + (100*1/10) data pages + (# tuples) = 1 + 10 + 1000 = 1011.

R join_a=b S join_b=c T

R(a):

- 1000 tuples; 100 pgs
- Primary tree index
- R.a uniform: 0-1000

S(b)

- 500 tuples: 50 pgs
- Primary heap file
- Secondary hash index on b
- S.b uniform: 0-50
- 1 bucket per S.b value

T(c)

- 10000 tuples: 1000 pgs
- Primary tree index
- Secondary tree index
- T.c uniform: 0-10
- ICARD(T.c) = 10

Tuples: 100bytes
Page: 1000 bytes
DE: 10 bytes
Fill factor: 100%
Tuples/page: 10
DE/page: 100

Cost copied from previous slide

R(a) index:

- R(a = ?): 2

S(b) secondary index:

- S(b=?): 11 pages.

T(c) primary index:

- T(c=?): 2 dir pages + (1000*1/10) = 102 pages

T(c) secondary index:

- T(c=?): 1 + (100*1/10) data pages + (# tuples) = 1 + 10 + 1000 = 1011.

Bestjoin(R join S):

R join S.

- R NL S: 100 + 100* 50 = 5100 pages
- R INL S: 100 + 1000 * 11 = 11100 pages
- R HJ S: 50 + 100 + 1000 * (10) = 10000 + 150 = 10150

S join R

- S NL R: 50 + 50 * 100 = 5050
- S INL R: 50 + 500 * 2 = 1000 + 50 = 1050
- **S HJ R: 100 + 50 + 500 * 1 = 650**

Bestjoin(S, T)
Bestjoin(R, T)
Bestjoin(Bestjoin(R, S), T)
Bestjoin(Bestjoin(S, T), R)
Bestjoin(Bestjoin(R, T), S)

R join_a=b S join_b=c T

R(a):

- 1000 tuples; 100 pgs
- Primary tree index
- R.a uniform: 0-1000

S(b)

- 500 tuples: 50 pgs
- Primary heap file
- Secondary hash index on b
- S.b uniform: 0-50
- 1 bucket per S.b value

T(c)

- 10000 tuples: 1000 pgs
- Primary tree index
- Secondary tree index
- T.c uniform: 0-10
- ICARD(T.c) = 10

Tuples: 100bytes
Page: 1000 bytes
DE: 10 bytes
Fill factor: 100%
Tuples/page: 10
DE/page: 100

Cost copied from previous slide

R(a) index:

- R(a = ?): 2

S(b) secondary index:

- S(b=?): 11 pages.

T(c) primary index:

- T(c=?): 2 dir pages + (1000*1/10) = 102 pages

T(c) secondary index:

- T(c=?): 1 + (100*1/10) data pages + (# tuples) = 1 + 10 + 1000 = 1011.

Bestjoin(R join S):

- **S HJ R: 100 + 50 + 500 * 1 = 650**

Cardinality of R join S?

* 1000 * 500 * (1/max(1000,50)) = 500 tuples = 50 pages

Bestjoin(Bestjoin(R, S), T):

- **SQ NL T: 650 + 50 * 1000 = 50000 + 650 = 50650**
- SQ INL T: 650 + 500 * 102 = 51000 + 650 = 51650
- SQ HJ T: 1000 + 650 + 500 * 1000 = 500000 + 1650 = 501650

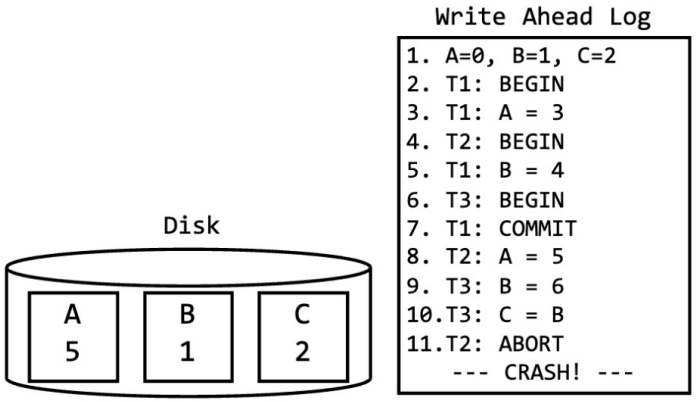
Analysis:
T1: committed
T2: abort
T3: abort

Which log records are reflected on the data pages on disk? And which need to be redone?
3: no redo
5: redo
8: no redo
9: redo
10: redo

Redo: 5, 9, 10
Undo: 10, 9, 8

4 (10 points) Recovery

The database contains three tuples A=0, B=1, C=2, and each page contains the corresponding tuple. T1, T2, and T3 start executing, and the database follows the write-ahead logging protocol as described in class. Each log record has an ID. The database crashes, and when the daabase recovers, it find the following contents of the disk and write-ahead log.



1. **(3 Points)** During the REDO phase of recovery, which page-write log records need to be re-executed? Fully fill in the circles next to those log record ids on the answer sheet.

2. **(3 Points)** During the UNDO phase of recovery, which page-write log records need to be undone? Fully fill in the circles next to those log record ids on the answer sheet.

1 ok
2 ok
3 ok
4 ok
5 ok
6 ok
7 ok
8. no

3. **(4 Points)** Consider the following sequence of database actions for one transaction. Check the statements where, if a crash occurs immediately after the statement, the database cannot correctly recover.

```
1. write B=2 in memory
2. write A=1 in memory
3. flush log record for "Write(A=1)"
4. write C=3 in memory
5. flush log record for "Write(C=3)"
6. flush C
7. flush A
8. flush log record for "COMMIT"
```

3. **(4 Points)** Consider the following sequence of database actions for one transaction. Check the statements where, if a crash occurs immediately after the statement, the database cannot correctly recover.

```
1. write B=2 in memory
2. write A=1 in memory
3. flush log record for "Write(A=1)"
4. write C=3 in memory
6. flush C
```

unsafe

```
5. flush log record for "Write(C=3)"
```